DESIGN AND DEVELOPMENT OF CUSTOMIZED COMMUNICATION PROTOCOL FOR THE SERIAL LINK INTERFACE FROM PCI-E ADD-ON CARD

D Anandaraj[1], Rajalakshmy sivaramakrishnan[2], G Karun kumar[3]
Flosolver unit, CSIR-NAL, Bangalore, Karnataka, India
Email : dar@flomail.nal.res.in[1], rsk@flomail.nal.res.in[2], karun.gorantla@gmail.com[3]

Abstract

This paper describes the design and development of the communication protocol for the customized data communication switch (FloSwitch). This protocol concentrates on the serial link interface between PCI-e add-on cards and the FloSwitch in the parallel computing system. The serial link is an optical interconnectivity with a bit rate of 6.25Gbps in full duplex mode. The protocol receives data from the PCI-Express (PCI-e) add-on cards through the optical link at the FloSwitch and does the necessary processing on the data (example: floating point addition), and the result is transmitted through the optical link to the destinations. This task is performed according to the command field, other information available in the header and the data. This implementation is done on the Field Programmable Gate Array (FPGA).

Keywords :
FloSwitch, PCI-e add-on card, optical link, floating point addition.

1. INTRODUCTION

The Flosolver unit of CSIR-NAL was initiated to build parallel computing systems. Towards this direction development was started in the year 1986 and India's first parallel computer Flosolver Mk1 was developed in 1986 [1]. Since then eight versions of parallel computers is developed from Flosolver Mk1 to Flosolver Mk8. Customized communication systems were designed, developed with various data communication devices and used in the different versions of parallel computing systems. This communication system is pluggable to any standard server with a PCI or PCI-e bus in it.

The latest Flosolver Mk8 parallel computer uses a customized communication system having PCI add-on cards and FPGA based FloSwitch. The Flosolver Mk8 organized as clusters of 128 modules. Each module consists of four Intel server boards with dual processor and a PCI add-on card which in turn interfaced to a FloSwitch. 128 such modules are interconnected to form a 1024 processors system. The PCI add-on card is a 64-bit/66MHz parallel interface, with a peak performance of 528MBps.

The FloSwitch is based on Virtex-5 FPGA having four parallel connectors for connecting the servers through PCI and 16 serial ports with optical transceivers for inter-connecting the different modules. Each transceiver supports a transfer rate of 6.25Gbps in full duplex mode. The communication within the module takes place through FloSwitch via the parallel interface and across the modules through the serial link - an optical cable. The FloSwitch is reconfigurable, and is designed for local and global communication [4]. The FloSwitch technology has a unique feature, a message processing capability, while message passing [7] and is designed to be efficient for operations such as global summation and global maxima calculations which are essential components of the parallel computing for meteorological applications.

The PCI add-on card permits short length of the parallel interface cable and lower data transfer rate. To enhance this, a PCI-e add-on card with 8 lanes is designed to support both 64bit parallel and 4 serial external interfaces. The PCI-e local bus is full duplex serial bus protocol operating at 2.5Gbps/lane/direction [2], and gives a throughput of 20Gbps/direction. The serial link and optical interconnect, with lower EMI [3], higher data transfer rate and greater cable length can be achieved.

This paper describes the design & development of the customized communication protocol for the FloSwitch for the serial link interface (optical link) from the PCI-e card [5].

2 DESCRIPTION

A communication protocol is designed, developed and implemented on the FloSwitch, for receiving the data from PCI-e card through the optical link, to perform the required operation and to transmit the result to the destination.

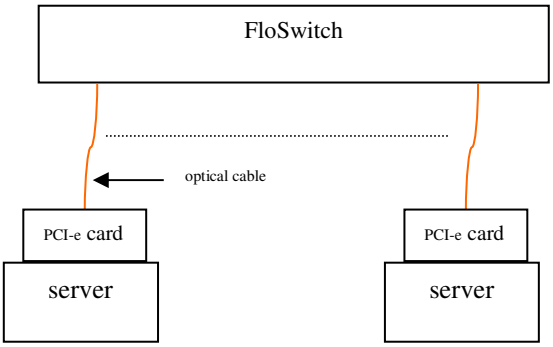


Fig 1. Servers connected to FloSwitch

The hardware setup is as shown in the Fig 1. The Intel server consists of 64bit/66MHz PCI & 8 lane PCI-e bus, with a Linux operating system on it. The device driver code and the custom Flo_MPI protocol, for the in-house built PCI/PCI-e card are implemented on to the operating system. The 8-lane PCI-e add-on card which consists of both 64bit parallel and 4 serial external interfaces is designed to interface to the FloSwitch. For the serial link, the optical cable is used as shown in Fig 1. Here the PCI-e card with an optical link is interfaced to the FloSwitch. The source and destination are the optical links.

The protocol implementation consists of two parts, (i) on server - implemented in C language and (ii) on FloSwitch - implemented in HDL and embedded C language. The protocol header information contains the service command along with the other information. The header & data which is sent by the server through PCI-e card via optical cable is received by the FloSwitch into the receiver buffer - RX-FIFO of the corresponding optical link in the FPGA. According to the service command, the FloSwitch performs the required operation on the data and then transmits the resultant back to the specified optical link/links according to the destination given in the header [6].

3 COMMUNICATION PROTOCOL

The protocol implementation is done on the server and the FloSwitch level. This report concentrates on the protocol for floating point addition along with the transfer of data.

In the server the following steps are carried out,

i. The floating point data and header are prepared, and then both are transmitted across the optical link as a single packet

ii. Then the server waits for the resultant data from the FloSwitch, by checking the buffer status of the particular optical link.

iii. Once the buffer status is updated, informing that the resultant data is available in the RX-FIFO, then the server receives the data

The FloSwitch implementation is done on the FPGA, having two parts and is implemented in Hardware Description Language (HDL) for 'hardware' and embedded C language for 'software'. The data and the header are received by the FloSwitch through the optical link, and then the content of the header is read. It then checked for the particular field in the header which differentiates between the parallel & serial interface. If that field contains a particular value, it indicates that the data & header is from the PCI-e serial link interface, otherwise it is for the parallel interface.

The following steps are carried out for the serial link interface.

i. The buffer memory BRAM has to be filled with zero data, to avoid any unwanted data added to the resultant.

ii. While processing the data, the content of the RX-FIFO and the BRAM is added and the result is stored back in the BRAM. If more servers are involved the above operation is repeated. The resultant data from BRAM is transmitted to the destination optical link/links.

3.1 IMPLEMENTATION

3.1.1 Header Structure:

The server initiates the service command. The header structure is prepared as given below. The same header which is received from server will

be transmitted back along with the resultant data. The PCIe_CMD, is a service command, which indicates what type of service the FloSwitch has to provide, the PCIe_ID field which differentiates between the parallel & serial link interfaces.

header{
  source cluster id
  source cpu id
  destination cluster id
  destination cpu id
  source id
  destination id
  data count
  number of servers involved
  processor_id[1-4]
  PCIe_CMD
  PCIe_ID
  Data [ ]
}

3.1.2 FloSwitch Implementation:

The FloSwitch implementation is done on the FPGA, having two parts and is implemented in Hardware Description Language (HDL) for 'hardware' and embedded C language for 'software'. The FloSwitch looks for the service command; once it receives the command the following tasks are performed. The following steps are carried out for the floating point addition.

i. The BRAM is filled with zero data. Then the DPU (Data Processing Unit) is configured based on the number of operands given in the header. A marking is made for the corresponding servers involved, into a register. Once the marking is completed for the involved servers, the task is performed.

ii. The data is added with the content of BRAM and the result is stored back in the BRAM. The operation is repeated for the number of servers involved. The number of servers involved can be random, but the operation will be performed according to the list. The resultant data from BRAM is transmitted to the destination optical link/links.

iii. Configure the CTU (Control & Timing Unit), with the information from the header. Then the following steps are carried out by the HDL,

- Reads the data from the RX_FIFO in order of the list
- The first packet (size is 1 to 1008 dwords) of data is added with the content of the BRAM
- The result is stored back in the BRAM
- The addition is repeated according to the list and the result is stored back in the BRAM
- A Interrupt is generated, indicating the processing is complete

iv. Once the Interrupt is generated, the following steps are carried out for transmitting the resultant data to destination optical link/links.

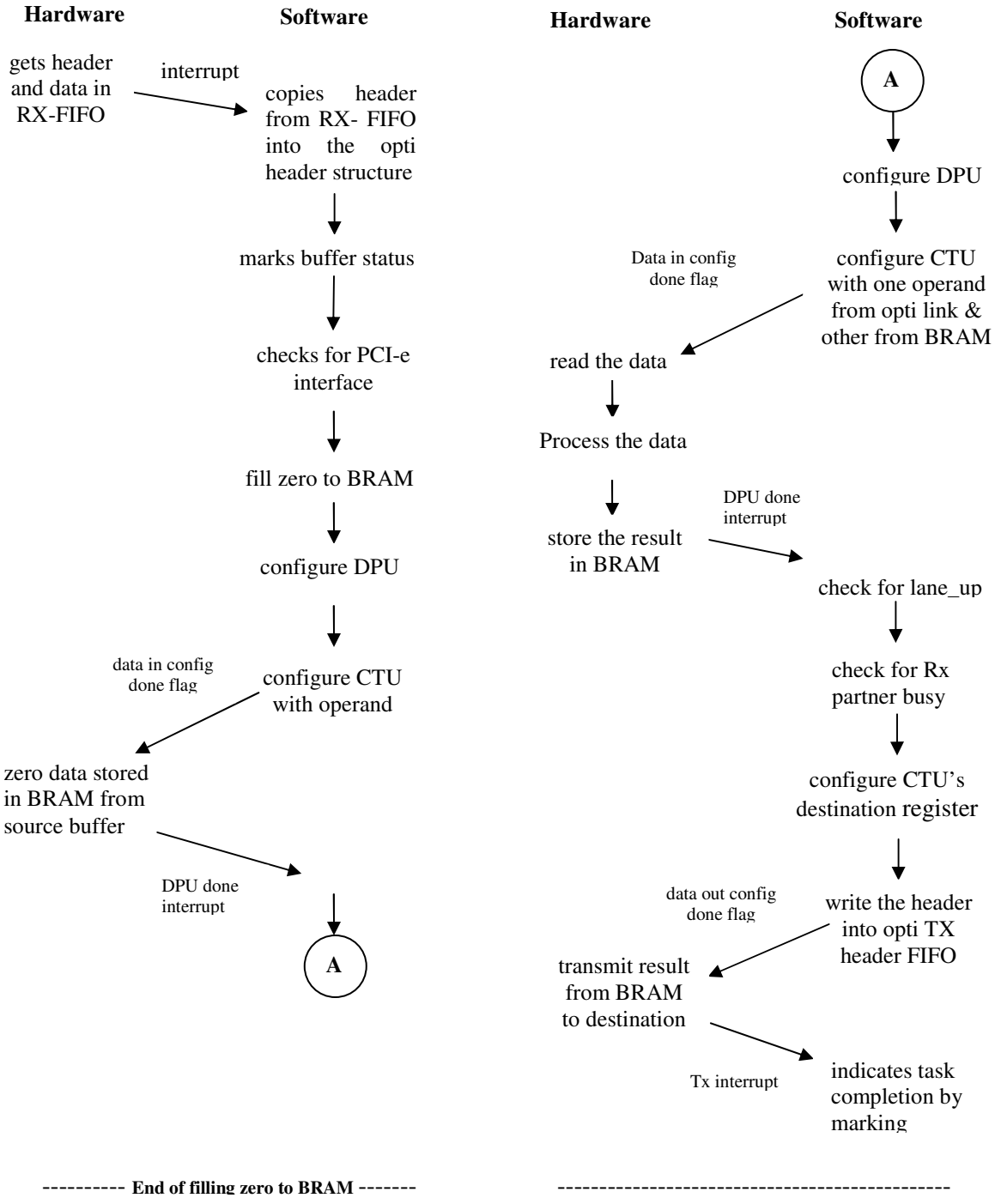
- Checks for whether optical lane is up
- Checks for receiver partner busy
- If the partner is not busy, then it transmits the resultant data according to the information from the header. The destination id field indicates the optical link/links, the data count indicates number of data which has to be transmitted. Along with the resultant data the header information is also transferred, so that the destination gets all the information about the data.
- TX_Interrupt is generated after the transfer is done, indicating the end of the transfer.

The flow diagram of the implementation is given in section 4. The flow consists of HDL for 'hardware' and embedded C language for 'software' in the FPGA.

The connectivity can be flexible within the module and across the modules as the optical interface having greater cable length, without any signal loss.

The communication protocol has been developed for the serial link and it has been tested successfully.

4. FLOW DIAGRAM



---------- **End of filling zero to BRAM** -------

---

5. CONCLUSION

The PCI-e card with an optical link is interfaced to the FloSwitch. Here the source & destination are optical links. The customized data communication protocol for the FloSwitch is designed, developed and implemented to support this serial link interface.

The preliminary testing is carried out and the test cases ran successfully. A 2.5Gbps bit transfer rate is achieved; where in the serial link with an optical interconnectivity supports a maximum bit rate of 3.125Gbps per direction.

The serial link with optical interface supports higher data transfer rate, and can have greater cable length without any signal loss, which leads to flexible inter-connectivity across the network. It also has lower EMI which ensures data strength.

REFERENCES

[1] U N Sinha, M D Deshpande, V R Sarasamma "Flosolver: A Parallel Computer for fluid dynamics", Current science, Volume 57, Pages 1277-1285, December 1988

[2] Ravi Budruk, Don Anderson, Tom Shanley, "PCI Express system Architecture", Mindshare

[3] Abhijit Athavale and Carl Christensen, "High-Speed Serial I/O Made Simple, A Designers' Guide, with FPGA Applications", Xilinx

[4] Flosolver Team, CSIR-NAL, "A study of FPGA modules on FPGA based FloSwitch", Project Document - PDFS1009 May2010, Flosolver Unit, CSIR-NAL

[5] Anirudh, Abhishek, Rizwan, Mainak Saha, Nirmal, Anandaraj, Venkatesh, Jagannadham, "Design of Flosolver PCI EXPRESS Card", Project Document - PDFS1013 September2010 Flosolver Unit, CSIR-NAL

[6] Anandaraj D, Karun kumar G and Swetha L, "Simulation of PCI-e card Optical interface to the FloSwitch", Project Document-PDFS1208 June2012, Flosolver Unit, CSIR-NAL

[7] U N Sinha, V R Sarasamma, Rajalakshmy Sivaramakrishnan, T N Venkatesh "A Device for scalable inter-nodal communication in a parallel computing system". Indian Patent No. 208824, 2007.